

Chatroom

Generated by Doxygen 1.10.0

1 File Index	1
1.1 File List	1
2 File Documentation	3
2.1 client.cpp File Reference	3
2.1.1 Macro Definition Documentation	3
2.1.1.1 BOLDWHITE	3
2.1.1.2 RESET	4
2.1.2 Function Documentation	4
2.1.2.1 main()	4
2.1.2.2 read_response()	4
2.1.2.3 receive_messages()	4
2.1.3 Variable Documentation	5
2.1.3.1 exit_flag	5
2.2 server.cpp File Reference	5
2.2.1 Macro Definition Documentation	5
2.2.1.1 BOLDWHITE	5
2.2.1.2 RESET	6
2.2.2 Enumeration Type Documentation	6
2.2.2.1 broadcast_type	6
2.2.3 Function Documentation	6
2.2.3.1 broadcast()	6
2.2.3.2 handle_client()	6
2.2.3.3 handle_disconnect()	7
2.2.3.4 main()	7
2.2.4 Variable Documentation	7
2.2.4.1 client_count	7
2.2.4.2 clients	7
2.2.4.3 mtx	7
Index	9

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

client.cpp	3
server.cpp	5

Chapter 2

File Documentation

2.1 client.cpp File Reference

```
#include <boost/asio.hpp>
#include <iostream>
#include <thread>
```

Macros

- `#define BOLDWHITE "\033[1m\033[37m"`
- `#define RESET "\033[0m"`

Functions

- string `read_response` (tcp::socket &socket)
Read a response from the server.
- void `receive_messages` (tcp::socket &socket)
Wait for messages from the server.
- int `main` ()
Main function.

Variables

- bool `exit_flag` = false

2.1.1 Macro Definition Documentation

2.1.1.1 BOLDWHITE

```
#define BOLDWHITE "\033[1m\033[37m"
```

2.1.1.2 RESET

```
#define RESET "\033[0m"
```

2.1.2 Function Documentation

2.1.2.1 main()

```
int main ( )
```

Main function.

Connect to the server and handle user input

Returns

int

2.1.2.2 read_response()

```
string read_response (
    tcp::socket & socket )
```

Read a response from the server.

Parameters

<i>socket</i>	The socket to read from
---------------	-------------------------

Returns

string The response from the server

2.1.2.3 receive_messages()

```
void receive_messages (
    tcp::socket & socket )
```

Wait for messages from the server.

Parameters

<i>socket</i>	The socket to receive messages from
---------------	-------------------------------------

2.1.3 Variable Documentation

2.1.3.1 exit_flag

```
bool exit_flag = false
```

2.2 server.cpp File Reference

```
#include <boost/asio.hpp>
#include <iostream>
#include <map>
#include <mutex>
#include <string>
#include <thread>
```

Macros

- `#define BOLDWHITE "\033[1m\033[37m"`
- `#define RESET "\033[0m"`

Enumerations

- enum `broadcast_type` { `Message` , `Announcement` }
- Broadcast types.*

Functions

- void `handle_disconnect` (const string &client_name)
Handle client disconnects.
- void `broadcast` (const string &data, const string &sender, `broadcast_type` broadcast_type)
Broadcast a message to connected clients.
- void `handle_client` (tcp::socket *socket)
Handle client connections.
- int `main` ()
Main function.

Variables

- std::map< string, tcp::socket * > `clients`
- std::mutex `mtx`
- int `client_count` = 0

2.2.1 Macro Definition Documentation

2.2.1.1 BOLDWHITE

```
#define BOLDWHITE "\033[1m\033[37m"
```

2.2.1.2 RESET

```
#define RESET "\033[0m"
```

2.2.2 Enumeration Type Documentation

2.2.2.1 broadcast_type

```
enum broadcast_type
```

Broadcast types.

Enumerator

Message	
Announcement	

2.2.3 Function Documentation

2.2.3.1 broadcast()

```
void broadcast (
    const string & data,
    const string & sender,
    broadcast_type broadcast_type )
```

Broadcast a message to connected clients.

Format message data and forward it to appropriate clients

Parameters

<i>data</i>	The data to be broadcasted
<i>sender</i>	The name of the sender
<i>broadcast_type</i>	The type of broadcast

2.2.3.2 handle_client()

```
void handle_client (
    tcp::socket * socket )
```

Handle client connections.

Read client name and messages

Parameters

<i>socket</i>	The client socket
---------------	-------------------

2.2.3.3 handle_disconnect()

```
void handle_disconnect (
    const string & client_name )
```

Handle client disconnects.

Parameters

<i>client_name</i>	The name of the client
--------------------	------------------------

2.2.3.4 main()

```
int main ( )
```

Main function.

Accept clients and create a new thread for each

Returns

int

2.2.4 Variable Documentation

2.2.4.1 client_count

```
int client_count = 0
```

2.2.4.2 clients

```
std::map<string, tcp::socket*> clients
```

2.2.4.3 mtx

```
std::mutex mtx
```


Index

Announcement
server.cpp, [6](#)

BOLDWHITE
client.cpp, [3](#)
server.cpp, [5](#)

broadcast
server.cpp, [6](#)

broadcast_type
server.cpp, [6](#)

client.cpp, [3](#)
BOLDWHITE, [3](#)
exit_flag, [5](#)
main, [4](#)
read_response, [4](#)
receive_messages, [4](#)
RESET, [3](#)

client_count
server.cpp, [7](#)

clients
server.cpp, [7](#)

exit_flag
client.cpp, [5](#)

handle_client
server.cpp, [6](#)

handle_disconnect
server.cpp, [7](#)

main
client.cpp, [4](#)
server.cpp, [7](#)

Message
server.cpp, [6](#)

mtx
server.cpp, [7](#)

read_response
client.cpp, [4](#)

receive_messages
client.cpp, [4](#)

RESET
client.cpp, [3](#)
server.cpp, [5](#)

server.cpp, [5](#)
Announcement, [6](#)
BOLDWHITE, [5](#)
broadcast, [6](#)

broadcast_type, [6](#)
client_count, [7](#)
clients, [7](#)
handle_client, [6](#)
handle_disconnect, [7](#)
main, [7](#)
Message, [6](#)
mtx, [7](#)
RESET, [5](#)